CLAIMS

What is claimed is:

1. A copy engine comprising:

a first register to point to a first address;

a second register to point to a second address, wherein one of the first and second addresses is a source address and one is a destination address for data to be copied;

a control register, wherein the control register includes:

a count of the amount of memory space required by a copy operation,

an indication of the direction of the copy operation from the first address to the second address or from the second address to the first address, and

an indication of whether the first memory address is incremented or decremented.

2. A copy engine as claimed in claim 1, wherein the copy engine includes a locking mechanism for locking the copy engine during a copy operation.

3. A copy engine as claimed in claim 2, wherein the locking mechanism is locked by a write to the control register and unlocked when the copy operation completes.

4. A copy engine as claimed in claim 1, wherein a write to the second address triggers the copy operation during which the copy engine is in an active state.

5. A copy engine as claimed in claim 3, wherein an attempt to write to the control register while the locking mechanism is locked is retried until the current copy operation has completed.

6. A copy engine as claimed in claim 4, wherein an attempt to write to any register during the active state is retried until the current copy operation has completed.

7. A copy engine as claimed in claim 1, wherein the copy engine includes a serialisation mechanism in which a write is made to the control register of zero count.

8. A copy engine as claimed in claim 1, including multiple sets of the first, second and control registers.

9. A copy engine as claimed in claim 8, wherein each set of registers can carry out copy operations simultaneously and can be locked independently.

10. A copy engine as claimed in claim 1, wherein an area of unused memory beyond the registers is provided to accommodate a cache line write to the registers.

11. A computer system comprising:

a central processing unit in which firmware is stored;

memory in which data is stored;

a copy engine which acts as an interface between the firmware and the memory; and

wherein the copy engine is as claimed in claim 1.

12. A computer system as claimed in claim 11, wherein the computer system includes a write queue in which waiting copy operations will stack up behind a write of zero count and the waiting copy operations will execute once the write of zero count has completed.

13. A copy engine comprising:

a first register to point to a first address;

a second register to point to a second address, wherein one of the first and second addresses is a source address and one is a destination address for data to be copied;

a control register, wherein the control register controls the addresses of memory space;

a locking mechanism for locking the copy engine during a copy operation;

a serialisation mechanism in which a write of zeros is made to the control register;

wherein, if the copy engine is unlocked, the write of zeros will execute with no effect and, if the copy engine is locked, the write of zeros will be retried until the previous copy operation has completed and the write of zeros will then complete with no effect.

14. A copy engine as claimed in claim 13, wherein the control register includes a count of the amount of memory space required by a copy operation and the write of zeros includes a zero count.

15. A copy engine as claimed in claim 13, including multiple sets of the first, second and control registers, wherein each set of registers can be locked independently.

16. A method of data movement comprising, a copy engine:

maintaining a first register to point to a first address;

maintaining a second register to point to a second address, wherein one of the first and second addresses is a source address and one is a destination address for data to be copied;

using a control register to:

count the amount of memory space required by a copy operation,

indicate of the direction of the copy operation from the first address to the second address or from the second address to the first address, and

indicate whether the first memory address is incremented or decremented.

17. A method as claimed in claim 16, wherein the method includes locking the copy engine during a copy operation.

18. A method as claimed in claim 17, wherein locking is activated by a write to the control register and deactivated by completion of the copy operation.

19. A method as claimed in claim 16, wherein a write to the second address triggers the copy operation during which the copy engine is in an active state.

20. A method as claimed in claim 18, wherein an attempt to write to the control register when locking is activated is retried until the current copy operation has completed.

21. A method as claimed in claim 19, wherein an attempt to write to any register during the active state is retried until the current copy operation has completed.

22. A method as claimed in claim 16, wherein the method includes serialisation by making a write to the control register of zero count.

23. A method as claimed in claim 22, wherein waiting copy operations stack up behind a write of zero count and the waiting copy operations execute once the write of zero count has completed.

24. A method as claimed in claim 16, wherein there are multiple sets of first, second and control registers and each set of registers carries out copy operations simultaneously and is locked independently.

25. A method as claimed in claim 16, wherein the method is carried out by a copy engine.

26. A method as claimed in claim 16, wherein firmware allocates an area of memory as free memory space and initialises the first register to point to the end of the free memory pages in memory.

27. A method of data movement comprising:

maintaining a first register to point to a first address;

maintaining a second register to point to a second address, wherein one of the first and second addresses is a source address and one is a destination address for data to be copied;

using a control register to control a copy operation;

locking the set of the first, second and control registers during a copy operation;

serialising copy operations by making a write of zeros to the control register;

wherein, if the set of registers is unlocked, the write of zeros will execute with no effect and, if the set of registers is locked, the write of zeros will be retried until the current copy operation has completed and the write of zeros will then complete with no effect.

28. A method as claimed in claim 27, wherein waiting copy operations stack up behind a write of zeros and the waiting copy operations execute once the write of zeros has completed.